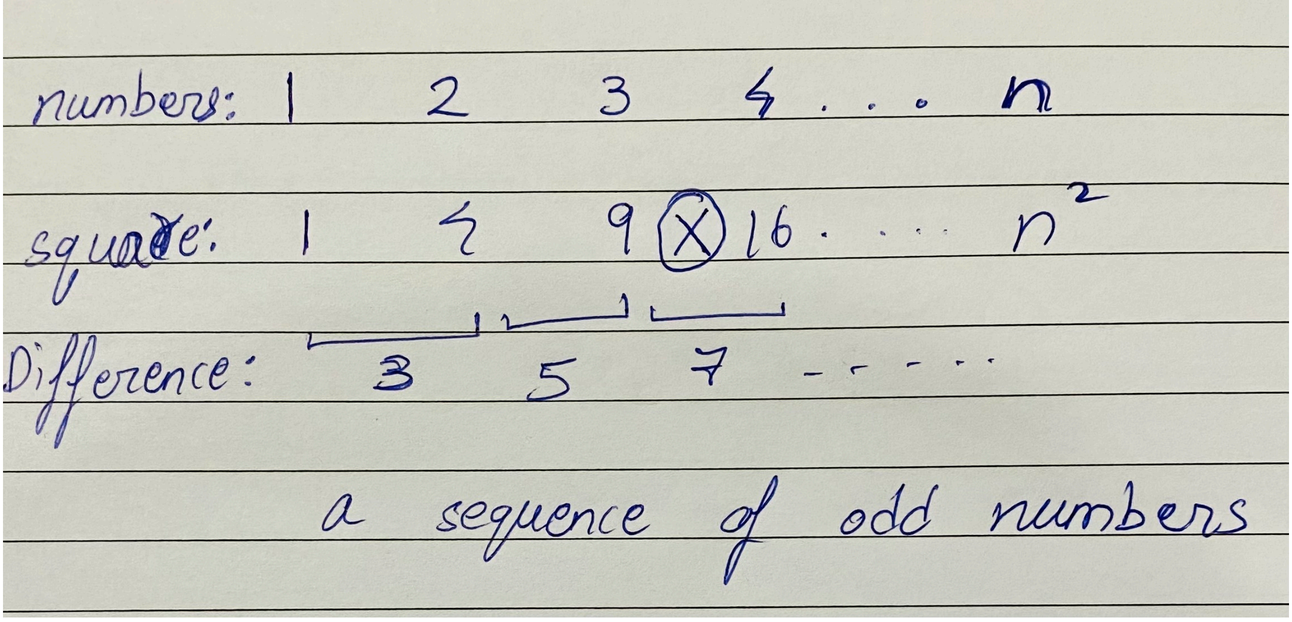
Digisim

Round 1

To design a digital circuit that takes 10 bit as input and gives 5 bit output which is square root of input number.

Way of approach:



numbers: 1 2 3 4 . . . n

square: 1 4 9 X 16 . . . n^2

Difference: 3 5 7 - - - - . .

a sequence of odd numbers

Our input number X lies between the squares or is one of them.
As the sum of n odd integers is n^2

1. We take input X
2. We start subtracting the odd numbers for input X. Until we get the output a -ve answer or Zero.
3. On another hand we cut the number of times we have subtracted an odd integer.
4. The final count of number of times we have subtracted the odd integer is our Square Root of X.

Working of circuit:

1. The input is supplied by user and is stored in D flip flop as initially the output value of the FSM is 0.
2. The circuit runs for one clock pulse and then value of FSM changes to 1 and stays 1. Hence triggering the 2nd army of buffer and blocking the 1st array of buffers, so is input by user.
3. The output of D flip flop is fed into an adder. The other input of adder comes form an odd counter. These values are subtracted if output is +ve the circuit will continue to run else it will stop and give output.
4. After every subtraction the output of Adders are fed to 2nd array of buffers which stores these values in the D flip flop.
5. The Counter counts the number of times the subtraction has took place, which is the result of square root of X.
6. This counter stops when the subtraction becomes -ve. Hence to stop the circuit we apply an AND Gate which has Cout and clock as input. To stop the output of AND is 0 hence not triggering the counter further.

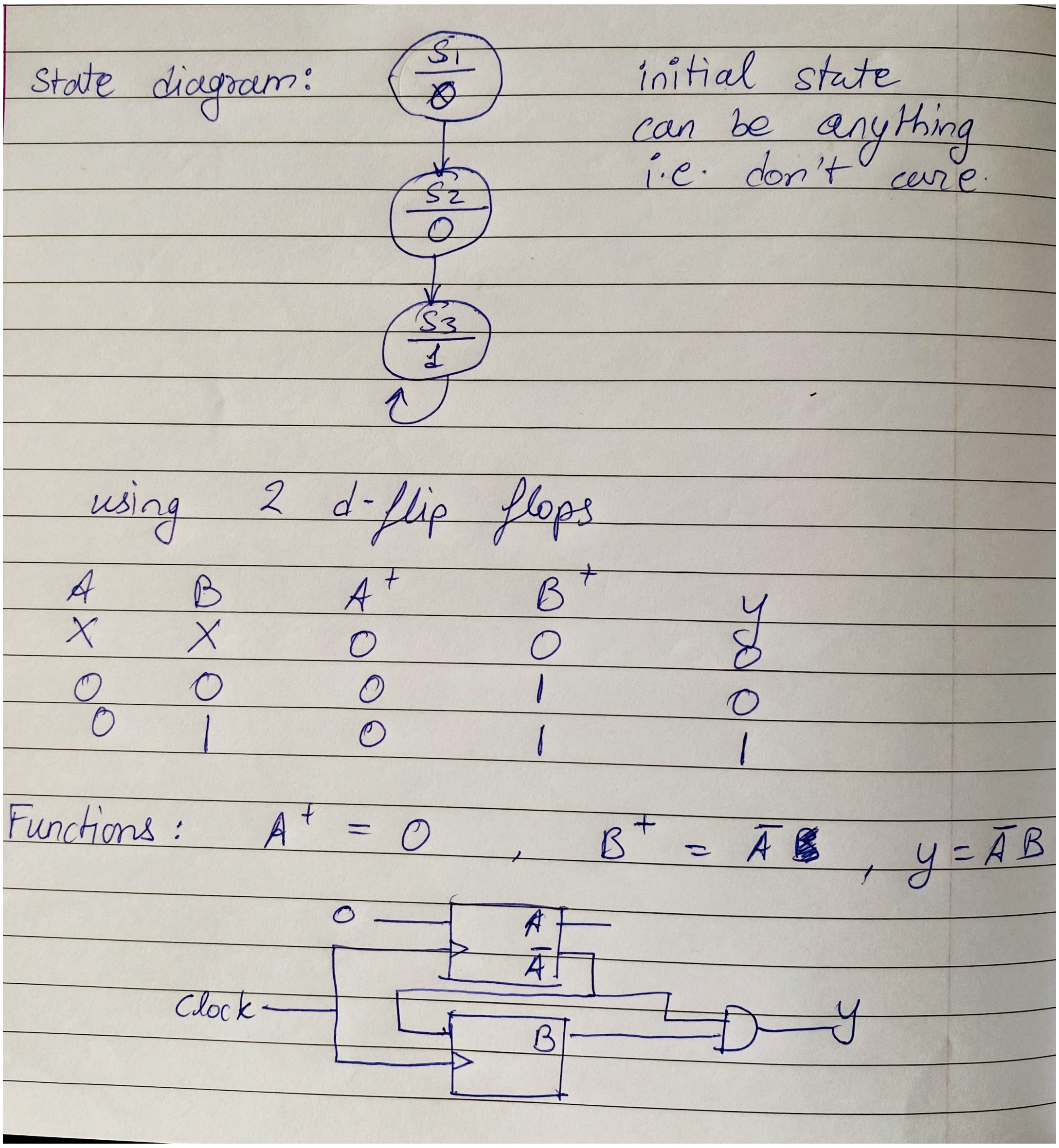
For following above given steps we use the following circuit and it is described as:

Taking input:
We take input by logic state. These are fed to an array of buffers which will allow the bit to pass only if the gate pulse is 0, else it gives output as high impedance.

As we want the input only once and rest of them it to be disconnected. We create another array of buffers whose inputs come from the adders and its gating pulse which is NOT of previous array. So only one of these array will be turned at a time.

The Gate value is given by the Carry Out of the last adder as it would give output 1, that is switching between arrays of buffers, when the Sum is -ve of input bits.

As we want the input given by user to be used for one pulse and for rest of time the circuit must be driven by the outputs of adders. For this we use a FSM design of Moore type.



State diagram:

S_1
X
S_2
0
S_3
1

initial state can be anything i.e. don't care.

using 2 d-flip flops

A	B	A^+	B^+	y
X	X	0	0	0
0	0	0	1	0
0	1	0	1	1

Functions : $A^+ = 0$, $B^+ = \bar{A}$ ~~B~~ , $y = \bar{A}B$

0
A
$\bar{A}$
Clock
B
y

We supply the the output of above FSM to arrays of buffers as it gives output 0 for just 1 clock pulse and 1 for rest of time.

D-flip flops for storing the inputs:
We supply the the output of the buffer array to a d flip flop and clock it to store the values in it and available to use further. The initial values X is stored in this flip flop and after. Also the Subtracted values of X that are outputs of adders.
The provided clock is of frequency f (explained in next point)

Counters and clock:
We are using 2 counter one for counting the odd numbers that we will supply into the adder for subtraction from the values stored in the D flip flops.

To get odd counter: We supply common clock to both the counters. However if we add a JK flip flop to one of the in beginning it becomes a counter with frequency f/2. This is reducing the frequency of clock. Now the clock with frequency 2f (clock directly supplied) will act as the Odd Number Counter. The counter with frequency f (cascaded with JKFF) will acts as counter which counts the number of times Subtraction occurs (giving answer).

Adders: Here they are used as **Subtractors**
The output of D flip flops storing the values of X is supplied to adder. And the values of counter of frequency 2f is supplied to adder by inverting them using not gate. Also the Cout for adder is 1 so that we are able to use 2's complement of number obtained by counter of frequency 2f.

The output of these adders are fed to the array pf buffers to store it in the D flip flop

Using AND Gate to fetch the results:
We use an AND Gate which has the clock as an input and the value of Cost of the adder as input. The Cout will turn to 0 when the output of adders is -ve and it is when we need to stop the circuit. So we use a not gate for this. When Cout=0 the output of AND Gate is 0 hence turning OFF the counters for further calculations as there will be no clock pulse.

Output:
For output also we have added buffers and Gating signal is Cout. So that the output is shown only when the whole process ENDS.

Component List:

Sr. no.	Component Name	IC Number	Quantity
1	Buffer	74125	25
2	D flip flop	74273	2
3	D flip flop	74175	1
4	Counters	74LS590	2
5	JK flip flop	74LS109	1
6	Adders	74283	3
7	Clock		1
8	NOT Gate		7
9	AND Gate		2

Team Name: V-Digit

Member 1: Ayush Singh Pal

Member 2: Astitva Gupta